

Python: module vcs.xvsy

vcs.xvsy

[index](#)

XvsY (GXY) module

Modules

[vcs.Canvas](#)
[vcs.VCS validation functions](#)

[vcs. vcs](#)
[cdtime](#)

[vcs.queries](#)

Classes

[builtin .object](#)
[GXY](#)

class **GXY**([builtin .object](#))

Class: [GXY](#) # XvsY

Description of GXY Class:

The XvsY graphics method displays a line plot from two 1D data arrays X(t) and Y(t), where t represents the 1D coordinate values. The class shows how to change line and marker attributes for the XvsY graphics.

This class is used to define an XvsY table entry used in VCS, or used to change some or all of the XvsY attributes in an existing entry.

Other Useful Functions:

a=vcs.init()	# Constructor
a.show('xvsy')	# Show predefined XvsY graph
a.show('line')	# Show predefined VCS line ob
a.show('marker')	# Show predefined VCS marker
a.setcolormap("AMIP")	# Change the VCS color map
a.xvsy(s1, s2, ,x,'default')	# Plot data 's1' and 's2' wi and 'default' template
a.update()	# Updates the VCS Canvas at
a.mode=1, or 0	If 1, then automatic update 0, then use update function Update the VCS Canvas.

Example of Use:

a=vcs.init()

To Create a new instance of XvsY use:

```

xy=a.createxvsys('new','quick')          # Copies content of 'quick'
xy=a.createxvsys('new')                  # Copies content of 'default'

To Modify an existing XvsY use:
xy=a.getxvsys('AMIP_psl')

xy.list()                                # Will list all the XvsY att
xy.projection='linear'                   # Can only be 'linear'
lon30={-180:'180W',-150:'150W',0:'Eq'}
xy.xticlabels1=lon30
xy.xticlabels2=lon30
xy.xticlabels(lon30, lon30)            # Will set them both
xy.xmtics1=''
xy.xmtics2=''
xy.xmtics(lon30, lon30)              # Will set them both
xy.yticlabels1=lat10
xy.yticlabels2=lat10
xy.yticlabels(lat10, lat10)           # Will set them both
xy.ymtics1=''
xy.ymtics2=''
xy.ymtics(lat10, lat10)              # Will set them both
xy.datawc_y1=-90.0
xy.datawc_y2=90.0
xy.datawc_x1=-180.0
xy.datawc_x2=180.0
xy.datawc(-90, 90, -180, 180)        # Will set them all
xaxisconvert='linear'
yaxisconvert='linear'
xy.xyscale('linear', 'area_wt')       # Will set them both

Specify the XvsY line type:
xy.line=0                                # same as xy.line = 'solid'
xy.line=1                                # same as xy.line = 'dash'
xy.line=2                                # same as xy.line = 'dot'
xy.line=3                                # same as xy.line = 'dash-dot'
xy.line=4                                # same as xy.line = 'long-dash'

Specify the Xvsy line color:
xy.linecolor=16    # color range: 16 to 230, default color is bl
xy.linewidth=16   # width range: 1 to 100, default color is bla

Specify the XvsY marker type:
xy.marker=1                                # Same as xy.marker='dot'
xy.marker=2                                # Same as xy.marker='plus'
xy.marker=3                                # Same as xy.marker='star'
xy.marker=4                                # Same as xy.marker='circle'
xy.marker=5                                # Same as xy.marker='cross'
xy.marker=6                                # Same as xy.marker='diamond'
xy.marker=7                                # Same as xy.marker='triangle'
xy.marker=8                                # Same as xy.marker='triangle'
xy.marker=9                                # Same as xy.marker='triangle'
xy.marker=10                               # Same as xy.marker='triangle'

```

```

xy.marker=11                      # Same as xy.marker='square'
xy.marker=12                      # Same as xy.marker='diamond'
xy.marker=13                      # Same as xy.marker='triangle'
xy.marker=14                      # Same as xy.marker='triangle'
xy.marker=15                      # Same as xy.marker='triangle'
xy.marker=16                      # Same as xy.marker='triangle'
xy.marker=17                      # Same as xy.marker='square'
xy.marker=None                     # Draw no markers

```

There are four possibilities for setting the marker color index

```

xy.markercolors=22                  # Same as below
xy.markercolors=(22)                # Same as below
xy.markercolors=([22])              # Will set the markers to a
                                    #      color index
xy.markercolors=None                # Color index defaults to Bla

```

To set the XvsY Marker size:

```

xy.markersize=5
xy.markersize=55
xy.markersize=100
xy.markersize=300
xy.markersize=None

```

Methods defined here:

```

__init__(self, parent, GXY_name=None, GXY_name_src='default', createGXY=0)

datawc(self, dsp1=1e+20, dsp2=1e+20, dsp3=1e+20, dsp4=1e+20)

list(self)

rename = renameGXY(self, old_name, new_name)
#####
#
# Function:      renameGXY
#
# Description of Function:
#      Private function that renames the name of an existing
#      graphics method.
#
#
# Example of Use:
#      renameGXY(old_name, new_name)
#          where: old_name is the current name of XvsY gr
#                  new_name is the new name for the XvsY g
#
#####

script(self, script_filename=None, mode=None)
    Function:      script                                # Calls _vcs.s

```

Description of Function:

Saves out a XvsY graphics method in Python or VCS script designated file.

Example of Use:

script(scriptfile_name, mode)

where: scriptfile_name is the output name of the
mode is either "w" for replace or "a" for

Note: If the filename has a ".py" at the end
Python script. If the filename has a ".scr"
produce a VCS script. If neither extension
default a Python script will be produced.

```
a=vcs.init()  
XY=a.createboxfill('temp')  
XY.script('filename.py') # Append to a Python file  
XY.script('filename.scr') # Append to a VCS file "f"
```

xmtics(self, xmt1=", xmt2="")

xticlabels(self, xtl1=", xtl2="")

yscale(self, xat=", yat="")

yntics(self, ymt1=", ymt2="")

yticlabels(self, ytl1=", ytl2="")

Properties defined here:

datawc_calendar

```
get">get = _getcalendar(self)  
set">set = _setcalendar(self, value)
```

datawc_timeunits

```
get">get = _gettimeunits(self)  
set">set = _settimeunits(self, value)
```

datawc_x1

```
get">get = _getdatawc_x1(self)  
set">set = _setdatawc_x1(self, value)
```

datawc_x2

```
get">get = _getdatawc_x2(self)  
set">set = _setdatawc_x2(self, value)
```

datawc_y1

```
get">get = _getdatawc_y1(self)  
set">set = _setdatawc_y1(self, value)
```

```

datawc_y2
    get">get = _getdatawc_y2(self)
    set">set = _setdatawc_y2(self, value)

line
    get">get = _getline(self)
    set">set = _setline(self, value)

linecolor
    get">get = _getlinecolor(self)
    set">set = _setlinecolor(self, value)

linewidth
    get">get = _getlinewidth(self)
    set">set = _setlinewidth(self, value)

marker
    get">get = _getmarker(self)
    set">set = _setmarker(self, value)

markercolor
    get">get = _getmarkercolor(self)
    set">set = _setmarkercolor(self, value)

markersize
    get">get = _getmarkersize(self)
    set">set = _setmarkersize(self, value)

name
    get">get = _getname(self)
    set">set = _setname(self, value)

projection
    get">get = _getprojection(self)
    set">set = _setprojection(self, value)

xaxisconvert
    get">get = _getxaxisconvert(self)
    set">set = _setxaxisconvert(self, value)

xmtics1
    get">get = _getxmtics1(self)
    set">set = _setxmtics1(self, value)

xmtics2
    get">get = _getxmtics2(self)
    set">set = _setxmtics2(self, value)

xticlabels1
    get">get = _getxticlabels1(self)
    set">set = _setxticlabels1(self, value)

```

```

xticlabels2
    get">get = _getxticlabels2(self)
    set">set = _setxticlabels2(self, value)

yaxisconvert
    get">get = _getyaxisconvert(self)
    set">set = _setyaxisconvert(self, value)

ymtics1
    get">get = _getymtics1(self)
    set">set = _setymtics1(self, value)

ymtics2
    get">get = _getymtics2(self)
    set">set = _setymtics2(self, value)

yticlabels1
    get">get = _getyticlabels1(self)
    set">set = _setyticlabels1(self, value)

yticlabels2
    get">get = _getyticlabels2(self)
    set">set = _setyticlabels2(self, value)

```

Data and other attributes defined here:

```

__slots__ = ['setmember', 'parent', 'name', 'g_name', 'yaxisconvert', 'xaxisconvert', 'linecolor', 'line',
'markersize', 'markercolor', 'projection', 'xticlabels1', 'xticlabels2', 'yticlabels1', 'yticlabels2', 'xmtics']

g_name = <member 'g_name' of 'GXY' objects>

parent = <member 'parent' of 'GXY' objects>

setmember = <member 'setmember' of 'GXY' objects>

```

Functions

```

getGXYmember(self, member)
#####
#
# Function:      getGXYmember
#
# Description of Function:
#     Private function that retrieves the XvsY members from the
#     structure and passes it back to Python.
#
#
# Example of Use:
#         return_value =

```

```

#           getGXYmember(self, name)
#               where: self is the class (e.g., GXY)
#                       name is the name of the member that is being
#
# #####
#
getmember = getGXYmember(self, member)
#####
#
# Function:      getGXYmember
#
# Description of Function:
#     Private function that retrieves the XvsY members from the
#     structure and passes it back to Python.
#
#
# Example of Use:
#     return_value =
#         getGXYmember(self, name)
#             where: self is the class (e.g., GXY)
#                     name is the name of the member that is being
#
# #####
#
renameGXY(self, old_name, new_name)
#####
#
# Function:      renameGXY
#
# Description of Function:
#     Private function that renames the name of an existing XvsY
#     graphics method.
#
#
# Example of Use:
#     renameGXY(old_name, new_name)
#         where: old_name is the current name of XvsY graphic
#                 new_name is the new name for the XvsY graphic
#
# #####
#
setGXYmember(self, member, value)
#####
#
# Function:      setGXYmember
#
# Description of Function:
#     Private function to update the VCS canvas plot. If the can
#     set to 0, then this function does nothing.
#
#
# Example of Use:

```

```

#           setGXYmember(self,name,value)
#
#               where: self is the class (e.g., GXY)
#                         name is the name of the member that is being
#                         value is the new value of the member (or att
#
#####
#setmember = setGXYmember(self, member, value)
#####
#
#   Function:      setGXYmember
#
#   Description of Function:
#       Private function to update the VCS canvas plot. If the can
#       set to 0, then this function does nothing.
#
#
#   Example of Use:
#       setGXYmember(self,name,value)
#               where: self is the class (e.g., GXY)
#                         name is the name of the member that is being
#                         value is the new value of the member (or att
#
#####

```

Data

StringTypes = (<type 'str'>, <type 'unicode'>)